Weather-Adaptive Windows

Team 27: Andy Lai, Michael Wu, Joseph Zatarski
TA: Michael Genovese

ECE 445 Final Report, Fall 2017

Abstract

We built a Weather-Adaptive Window that would respond to certain weather conditions and raise or lower a window pane appropriately. In the following report, we will introduce the motive and objective behind engineering our project. We will then give a high-level systems overview followed by a detailed look at the various modules of the window system, along with the steps we took to validate each subsystem. Finally, we will present the results of our project.

Contents

1 Introduction **1**
- 1.1 Objective . . . 1
- 1.2 Background . . . 1
- 1.3 High Level Requirements . . . 1

2 Design **2**
- 2.1 Power Module . . . 2
  - 2.1.1 Wall Adapter . . . 2
  - 2.1.2 DC-DC Step Down Regulators . . . 3
- 2.2 Control Module . . . 3
  - 2.2.1 Microcontroller . . . 4
  - 2.2.2 Wi-Fi Module . . . 5
- 2.3 Sensor Module . . . 6
  - 2.3.1 Rain Sensor . . . 6
  - 2.3.2 Temperature Sensor . . . 6
  - 2.3.3 IR Sensor . . . 7
- 2.4 Motor Drive Module . . . 7
  - 2.4.1 Full Bridge Motor Driver . . . 7
  - 2.4.2 Motor . . . 9
- 2.5 Physical Design . . . 10

3 Verification **12**
- 3.1 Power Module . . . 12
- 3.2 Control Module . . . 13
- 3.3 Sensor Module . . . 13
- 3.4 Motor Drive Module . . . 13

4 Cost **14**
- 4.1 Labor Cost . . . 14
- 4.2 Material Cost . . . 15

5 Conclusion **17**
- 5.1 Accomplishments . . . 17
- 5.2 Uncertainties . . . 17
- 5.3 Ethics and Safety . . . 17
- 5.4 Future Work . . . 18

A Schematics and Layouts **19**

B Requirements and Verification **23**

C Tolerance Analysis **27**

List of Figures

1 High Level Power Block Diagram 2
2 High Level Data Block Diagram 3
3 Power Supply Schematic 4
4 Window State Machine 5
5 One Half of the Full Bridge Driver 8
6 Current Sense Comparator Circuitry 10
7 Inside and Outside of Assembled Window 11
8 Ripple in Buck Converters During Motor Current Flip 12
9 Oscilloscope Readout of the ADC Output 14
10 Main Board Schematic Page 1: Microcontroller, Wi-Fi Module, Buck Converters 19
11 Main Board Schematic Page 2: Motor Drivers, Current Sense Comparators . . . 20
12 Main Board PCB Layout 20
13 Temperature Sensor Schematic 21
14 Temperature Sensor PCB Layout 21
15 Infrared Sensor Schematic 22
16 Infrared Sensor PCB Layout 22

1 Introduction

1.1 Objective

Most windows installed in buildings today are manual. Once left in a state, open or closed, by a user, there is nothing that can be done about the state of the window until another user manipulates the window. This can lead lead to some negative consequences. If a person leaves the window open at night to reduce the temperature of his or her bedroom, a constant breeze could lead to adverse health effects. If windows are left open during a storm, water could enter the building, causing potential damage and flooding. If a window is left open and the weather gets too hot or cold (relative to the desired temperature), keeping the window open could lead to the opposite of the desired result. Last, but certainly not least, there is always the factor of human error; simple forgetfulness can lead to a window being open or closed when it is not supposed to be. This is why it is important to have a window that controls itself based on a couple of important weather conditions - temperature and rain. In addition, it is also important to give the user remote control over the position of the window.

We built a Weather-Adaptive Window that also provides wireless user control via Wi-Fi. Sensors that monitor the ambient environment will alert the control system to inclement or undesired weather, and an open window will automatically shut itself. To protect homeowners as well as their pets and children, these windows will also respond to the presence of an obstacle and remain open. In the extremely unlikely event that the window does close on an object, the control circuitry should detect a current spike from the motor and shut down operations.

1.2 Background

The rapid rise of the home automation market, which was valued at $5.77bn in 2013 and is expected to grow at a CAGR of 11.36% between 2014 and 2020 [1], speaks to the increasing need of convenience and connectivity in people's lives. Nothing is more convenient than when something takes care of itself, and perhaps the second best option is to be able to control something from anywhere. A great example of this is the burgeoning Nest Learning Thermostat. The thermostat can be programmed via an app, which means that the temperature of one's house can be controlled from anywhere, potentially leading to massive power savings. After a week of use, the thermostat "learns" the typical behavior of the user and will begin programming itself [2]. Although the use of a smart thermostat can result in good power savings, even more drastic savings can be incurred by using the ambient outdoor temperature to regulate the temperature of indoor spaces. During mild seasons or in places with a temperate climate, being able to leverage this advantage is very important. Not only is the advantage in monetary savings, but it also reduces the carbon footprint of indoor spaces.

1.3 High Level Requirements

1. Window should be able to open and close itself based on a temperature threshold.
2. Window should close itself in the presence of rain.
3. User should be able to wirelessly control the window using their mobile device.

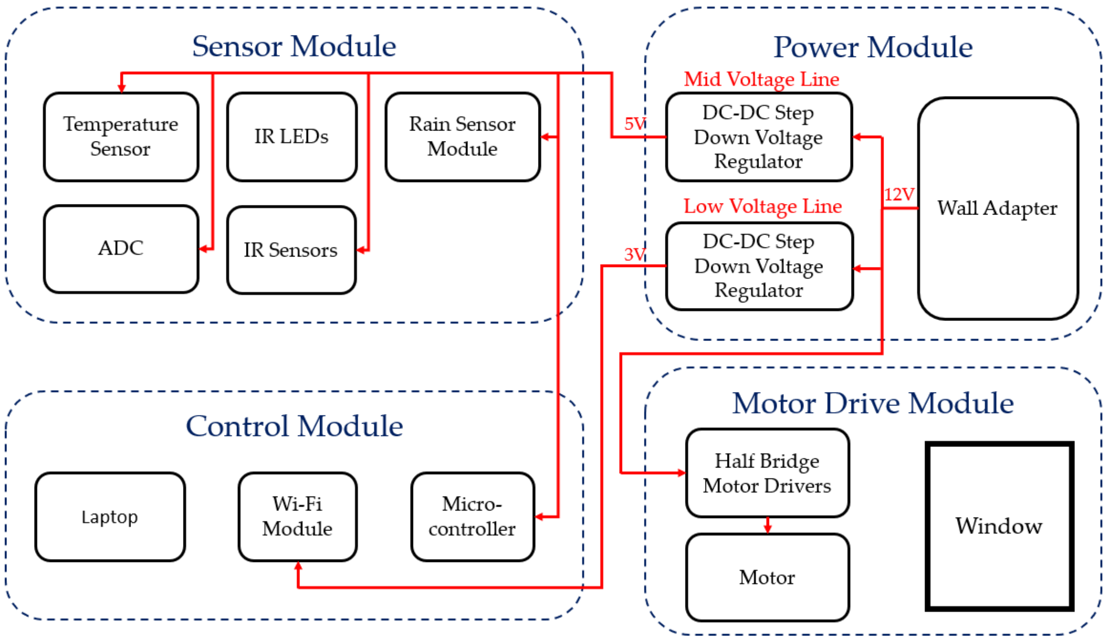


Figure 1: High Level Power Block Diagram

2 Design

The Weather-Adaptive Windows operates using four modules, shown in Fig. 1. The power module manages power supplied from standard American 60 Hz 120 V alternating current (AC) wall voltage so that the sensors are constantly operable. The sensor module receives feedback from the environment, processes the data where necessary, and communicates the information to the controller. The controller module involves an integrated Wi-Fi chip and RF development board to provide control for the Weather-Adaptive Window as well as user interface. Finally, the motor drive module receives input from the control module to open and close the window. Block diagrams can be seen in Fig. 1 and Fig. 2.

2.1 Power Module

The purpose of the Weather-Adaptive Window is that it will close in any inclement weather conditions, meaning the sensors must always be operating. For this reason, we choose to power the window and its components from the wall. A schematic can be found in Fig. 10.

2.1.1 Wall Adapter

The wall adapter was required to convert the voltage from alternating current (AC) to direct current (DC) and step it down to a manageable voltage. We selected a wall adapter that converts the voltage from 120 V, 60 Hz AC to 12 V DC and could achieve that with at least 80% efficiency. The output voltage was chosen in order to directly supply the motor with enough power while

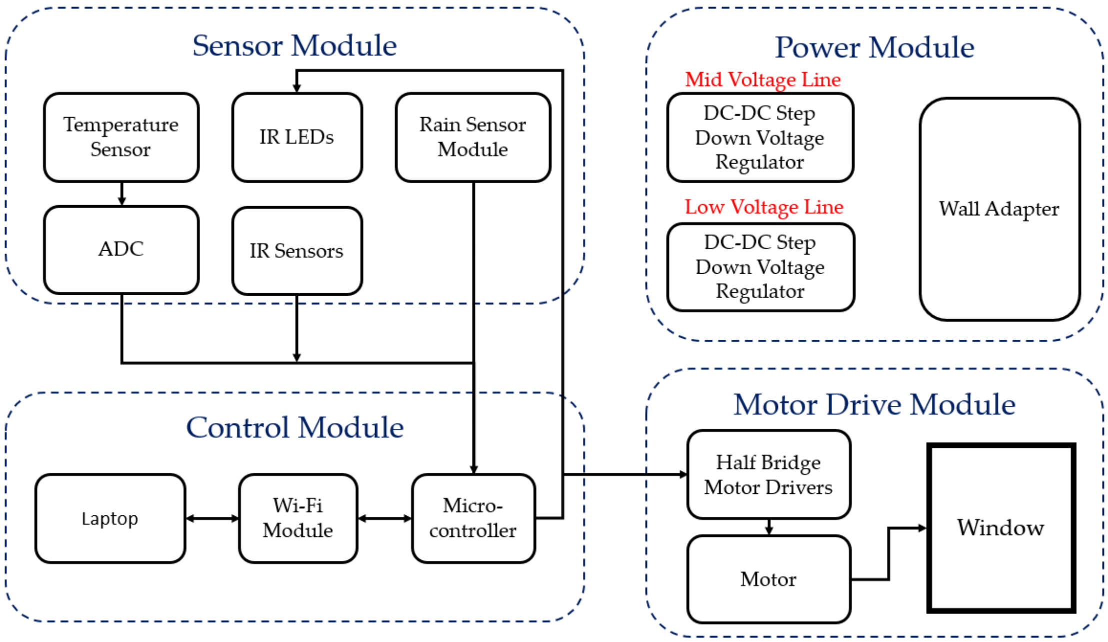


Figure 2: High Level Data Block Diagram

also providing a reasonable intermediate voltage that we could step down further down the power pipeline to power the other low voltage components of the circuitry. Because we were working with a motor, which drew a lot of current, we needed to select an adapter that could handle up to 5 A stall current, based on the specifications of the selected motor. To meet these requirements we chose a 96 W AC-DC converter, capable of sourcing 12 V and 8 A [4].

2.1.2 DC-DC Step Down Regulators

The DC-DC Step Down (buck) regulators were required to convert the voltage between intermediate rails, as the various components of the circuitry have different power requirements. Since the power came from the wall, reliability was more important than efficiency. In addition, we prioritized sensors that have low quiescent current, allowing us to impose stricter tolerance limits at the expense of efficiency. There were two low voltage power rails we needed to supply: a mid-level rail at 5 V, which supplies the microcontroller and sensors; and a low-level rail at 3.3 V, which supplies everything else. To meet these requirements we have chosen the TS30013-M033QFNR and TS30013-M050QFNR, respectively [5].

2.2 Control Module

The control module interfaces to the sensor module and the motor driver module to monitor the weather conditions and control the window mechanism. The control module also had Wi-Fi capability to allow remote control of the window through a smartphone application, but this wasn't implemented in time for the demonstration.

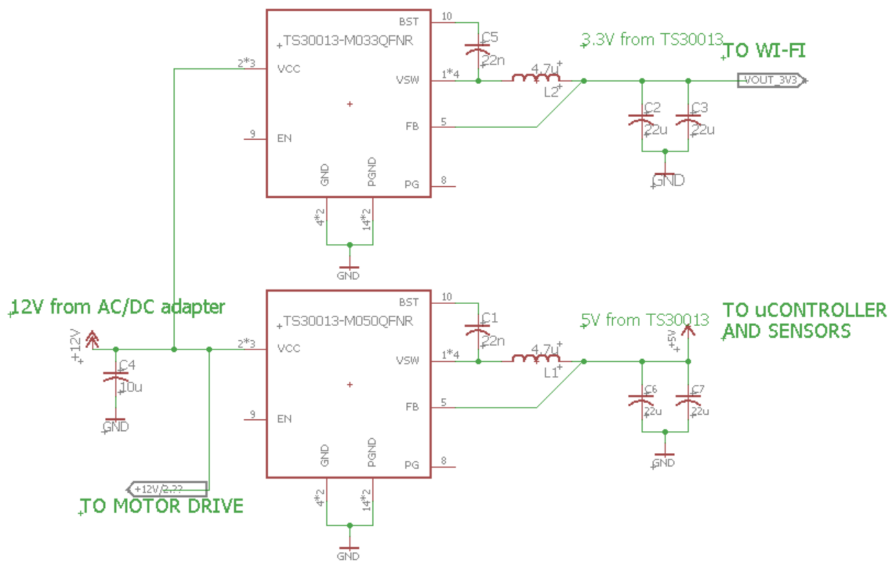


Figure 3: Power Supply Schematic

2.2.1 Microcontroller

The microcontroller hosts the software which controls the window based on the weather conditions read from sensor inputs. We chose to go with the ATmega328P [6], as its numerous capabilities and easy user-interface will made testing hardware and software compatibility straightforward. Some component requirements we considered while selecting the part included sufficient I/O capabilities to interface with the peripherals, which include the temperature sensors, IR sensors, rain sensor, motor driver, and Wi-Fi module. In addition, it was required that the microcontroller be able to poll the IR sensors quickly to prevent the window from closing down on an object in the way.

The microcontroller intelligently controls the window via a state machine, seen in Fig. 4. There exist four states: top, bottom, up, and down. During the top and bottom states, the motor is turned off; during the up and down states, outputs are set to make the motor move in the according directions. From the top state, it is only possible to transition to the down state if it is raining or via a temperature mismatch, i.e. it is hotter/colder than desired outdoors). However, if there is an object in the way of the window, the window will not close. From the down state, the down state can be reached once the jam detection on the bottom is triggered along with the bottom-sensing IR sensor. However, if an object blocks the window along the way, it will immediately bounce upward, much like a garage door. From the bottom state, it is only possible to transition to the up state via a different temperature mismatch, i.e. it is a desirable temperature outside. However, if it is raining, the window will not move from the bottom state. From the up state, only the top state can be reached by jam detection on the top.

The temperature mismatch is calculated by comparing, in software, the inside and outside ambient temperatures against a user preset value. If the indoor temperature can be changed to be closer to the desired temperature by letting outside air flow in, the window will respond

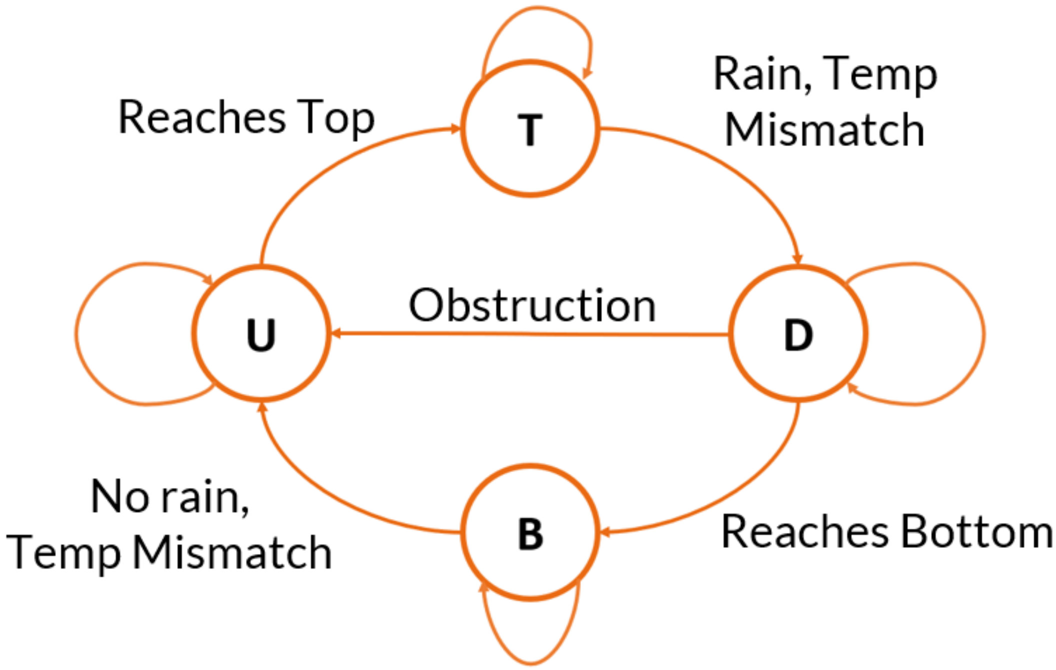


Figure 4: Window State Machine. The four states, top, bottom, up, and down, are denoted by T, B, U, and D respectively. Additionally, state transitions are labelled - any inputs that aren't labelled maintain the current state.

accordingly. For instance, if the indoor temperature is warmer than the desire temperature while the outdoor temperature is cooler, the window will respond by opening. These state transitions are overridden by rain and obstructions. To prevent the window from oscillating between states too quickly, we implement a pseudo-hysteresis by imposing a wide threshold around the current indoor temperature, keeping in mind that temperature typically changes very slowly.

2.2.2 Wi-Fi Module

Our Wi-Fi connectivity was provided by the ESP8266 [7] on the 8-pin ESP8266-01 module. This component has a PCB trace antenna capable of 2.4 GHz compliant with IEEE 802.11 standards. In addition, it is capable of serial communication with the microcontroller via a RX and TX line. During the design phase, we prototyped the ESP8266 module by using an Arduino Uno board as a proxy and programming through AT commands, and were able to connect to Wi-Fi, set up a TCP server, and communicate with a laptop. We chose this module as it is capable of being bootloaded with Arduino code; however, we fell short of fully implementing Wi-Fi capability.

2.3 Sensor Module

We designed the window using sensors that were in line with weather conditions that prompt most users to open and close their windows, namely rain and temperature. These sensors combine a small footprint with the ability to provide information that is immediately pertinent to current weather conditions. We felt that incorporating other weather conditions such as wind or barometric pressure either required a large and unwieldy sensor, or provided information beyond what the typical user requires to open or close a window.

2.3.1 Rain Sensor

The rain sensor is a printed circuit board (PCB) with exposed leads that act as a circuit that is completed in the presence of water. This PCB is connected to a module with an on-board voltage comparator, outputting a digital high to the microcontroller in the presence of rain. In addition, the module had two status LED's, one signifying operation and the other signifying the presence of water. We mounted the exposed-lead PCB on the horizontally on the outside of the window on an L-bracket, and kept the comparator on the inside.

The rain sensor operates at 5 V digital logic. This made interfacing with the power module and control module simple.

2.3.2 Temperature Sensor

The temperature sensor should monitors the outside temperature to within $\pm 1°$C. The analog output of the temperature sensor should feed through an analog to digital converter (ADC) which can be read in by the microcontroller. We would like to have low operating power temperature sensors, as they must be always listening. This trumps having a high-performance temperature sensor. So that our microcontroller will not need a built in ADC or an analog IO pin, we will use a serial output ADC and minimize the GPIO needs of the temperature sensor. We have chosen the MCP9700A [8] and the ADS7868 [9] for the temperature sensor and ADC, respectively.

The temperature sensor should monitors the outside temperature to within $\pm 1°$C. We chose the MCP9700A [8] for its minimal footprint and extremely low operating current. Our baseline is a typical user's perception of relative temperature which they would use to make decisions on the state of a window. As a human cannot be expected to estimate temperatures to within tenths of a degree, we felt that a small footprint to minimize sensor board space was far more important than extreme accuracy. The MCP9700A operates within -40 to $+125\circ$C, which is within typical living conditions.

In order to alleviate the need for an analog IO pin on our microcontroller, we chose to feed the output of the temperature sensor through a serial ADC. For this part we chose the 8-bit ADS7868 [9]. As the MCP9700A has a maximum output voltage of 1.75 V, we chose to power the ADC with 3.3 V to compromise between ease of interface with the 5 V microcontroller and finer resolution on the temperature sensing. We calculate the temperature resolution as follows, noting that the MCP9700A outputs a minimum of 0.1 V at -40°C and a maximum of 1.75 V at +125°C.

$$\frac{3.3\text{V}}{2^8} \times \frac{125°\text{C} + 40°\text{C}}{1.75\text{V} - 0.1\text{V}} = 1.29°\text{C} \tag{1}$$

With a resolution of 1.29°C, satisfy our accuracy requirement of ±1°C across a range of 165°C. To provide a 3.3 V reference voltage for the ADS7868, we used the REF3133 [10].

2.3.3 IR Sensor

The main purpose of IR sensors is to detect the presence of an obstruction in the path of the closing window, so that the motor can stop closing the window and instead raise it in the interest of safety. Secondarily, an IR sensor is also used to determine when the window has reached the bottom of the frame. Along with using the motor driver's current sense to detect that it has reached the bottom, this two-factor location determination means we can distinguish instances of the window striking the bottom of the frame from striking an object.

Three IR sensors are placed along the height of the window frame to detect obstructions, with an IR LED placed on the opposite side to construct a break beam sensor. We chose to use just one LED for all three sensors as increasing the number of emitters led to flooding, so no sensor could detect that a beam had been broken. The bottom-detect IR sensor and its accompanying emitter are also arranged as a break bream sensor; when the window is high, it breaks the beam, and the beam is connected when the window is at the bottom.

We used an ILED-8 [12] as the emitter and a TSSP77038 [11] as the sensor, both optimized at a 940nm wavelength. The TSSP77038 has a built-in bandpass filter optimized at 38 KHz, occluding ambient IR light to detect the desired pulsed signal. However, when we pulsed the IR LED at near 38 KHz, we found that it flooded the relatively short span of the window, and the sensors could not reliably detect on object. Therefore we lowered the pulse rate to 15.6 KHz. In addition, we built in over-voltage protection (OVP) for the IR sensors using the GSOT03C electrostatic discharge protection diodes due to possibly high transient voltages from motor operations.

The IR emitters were driven in software using the microcontroller's timer interrupt. This allowed the system to pulse the LED's in the background without affecting foreground operations. The microcontroller uses a timer computed by dividing the internal clock by a prescaler. Every time a preset value is reached by the timer, the timer is cleared and begins again. We used the following calculation to determine the desired preset value.

$$\text{Timer Val} = \frac{\text{Clock Speed}}{\text{Prescaler}}/(2 \times \text{Desired Frequency}) \tag{2}$$

Multiplying the desired frequency by 2 is necessary as each jump to the interrupt service routine toggles the state of the LED on or off; thus, we need to interrupt at twice the desired frequency. Choosing a prescaler of 256 and a timer comparator value of 2 yielded an operating frequency of 15.6 KHz.

$$\frac{16\text{MHz}}{256}/(2 \times 2) \approx 15.6\text{KHz} \tag{3}$$

2.4 Motor Drive Module

2.4.1 Full Bridge Motor Driver

The motor driver needed to be capable of driving a DC motor in two directions. One direction is for closing the window, and the other direction is for opening it. The two most common ways

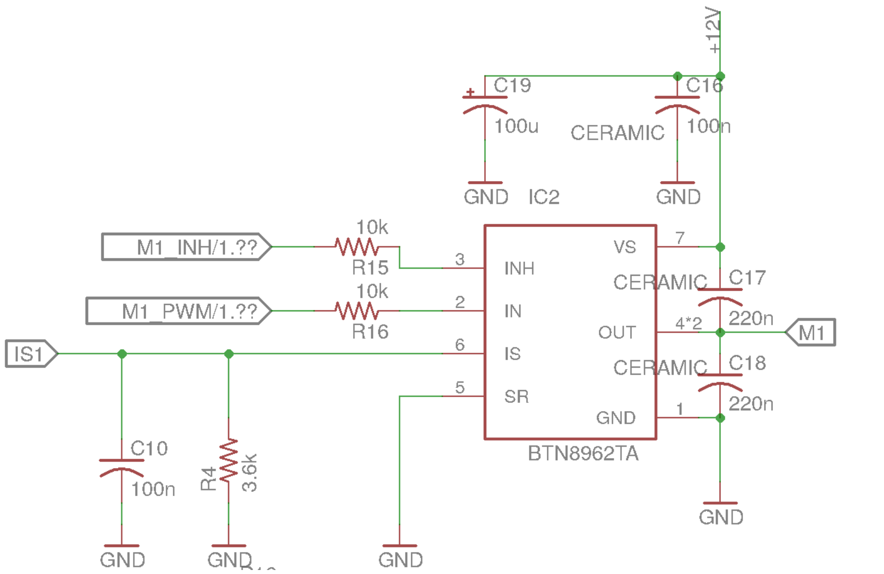


Figure 5: One Half of the Full Bridge Driver

of doing this are to use a full bridge with a single rail power supply, or to use a half bridge with a split rail power supply. The full bridge topology was chosen to simplify the power supply requirements. Additionally, the motor driver needed to have the capability to measure the motor current for feedback purposes. If the motor jams or the window is obstructed in some way, there will be a current spike due to the motor stalling. The microcontroller must stop the motor in this case, to prevent motor overheating. The full bridge driver was also designed with PWM control in mind due to concerns about motor inrush current, although this was later determined to be unnecessary.

To meet these requirements, we chose the BTN8962TA [14]. Two of these half bridge drivers were used to construct the full bridge. The circuitry for each of these half bridges is shown in figure 5. IC2 is the half bridge driver IC itself. C19 and C16 smooth the supply voltage from the motor current transients during switching. C17 and C18 protect the driver IC from high voltage switching transients produced by the motor during turn-on and turn-off. Lastly, R4 and C10 form the current feedback network. The BTN8962 outputs a current proportional to the current in the driver on the IS pin, which is converted into a voltage by R4. C10 acts as a low pass filtering high frequency noise from the output.

The value of R4 was carefully chosen so that the voltage did not exceed 5V during worst case motor current, but also to maximize dynamic range of the current sense output. The worst case current for the motor is 5 A at 12 V at stall. BTN8962TA's worst case current sense ratio is 7200.

$$\frac{5A}{7200} = 694\mu A = I_s \tag{4}$$

The worse case I_s offset is 440 μA. Therefore, the total worst case maximum sense current is

$$I_s = 694\mu A + 440\mu A = 1.13mA \tag{5}$$

To add some buffer,

$$1.2 * 1.13mA = 1.36mA \tag{6}$$

1.36mA should correspond to the maximum input of a voltage comparator. The current sense resistor should be chosen to produce 5V and no more.

$$\frac{5V}{1.36mA} = 3670\Omega = R_{max} \tag{7}$$

For a 5% resistor, $\frac{3670}{1.05} = R_{nom} = 3500\Omega$. The current sense resistor selected was $3600\Omega = 3.6k\Omega$, which is a standard E24 value. We will attempt to clamp I_s output to roughly 5V using a diode. Assuming that the voltage is clamped to 5.4V,

$$\frac{(5.4V)^2}{3.6k\Omega * 0.95} = 8.5mW \tag{8}$$

This means that a 0603 resistor should suffice.

The output of the current sense network is then fed to a comparator. The circuitry for this comparator is shown in figure 6. The comparator used is an LM393, an IC with two comparators on it. The comparators have an open collector output which means they require a pullup resistor on the output such as R7. R9 and R10 set the comparator hysteresis. Having a small amount of hysteresis prevents the comparator from oscillating in the presence of noisy input. A 100K potentiometer allows the comparator threshold to be set.

2.4.2 Motor

The motor operates the hung window frame via a lead screw and has the purpose of physically opening and closing the window. The motor has requisite torque, power, and speed to open and close the window at least a foot in roughly 15 seconds. To meet these requirements, we selected a metal gearmotor with a 50:1 gear reduction with 200 rotations per minute (RPM) and 300mA in free-run.

We begin by calculating requisite torque.

$$l \times TPI \times 2\pi \times \tau = F \times l \tag{9}$$

$$\tau = \frac{F}{TPI \times 2\pi} \tag{10}$$

where TPI stands for "Turns per Inch".

Supposing a 14TPI lead screw and a 50lb operating force, a huge overestimate, we have

$$\tau = \frac{50lb}{14TPI \times 2\pi[rads/turn]} = .568lb - in = 9.1oz - in \tag{11}$$

We then calculate requisite speed, assuming the window must move 12 inches to follow open and close.

$$\frac{12in}{1min} = \frac{RPM}{14TPI} \rightarrow 168RPM \tag{12}$$

To meet these requirements, we selected a metal gearmotor, a Pololu 12 V motor with a 50:1 gear reduction. The output speed with no load is 200 RPM, and the stall torque is 170 oz-in.

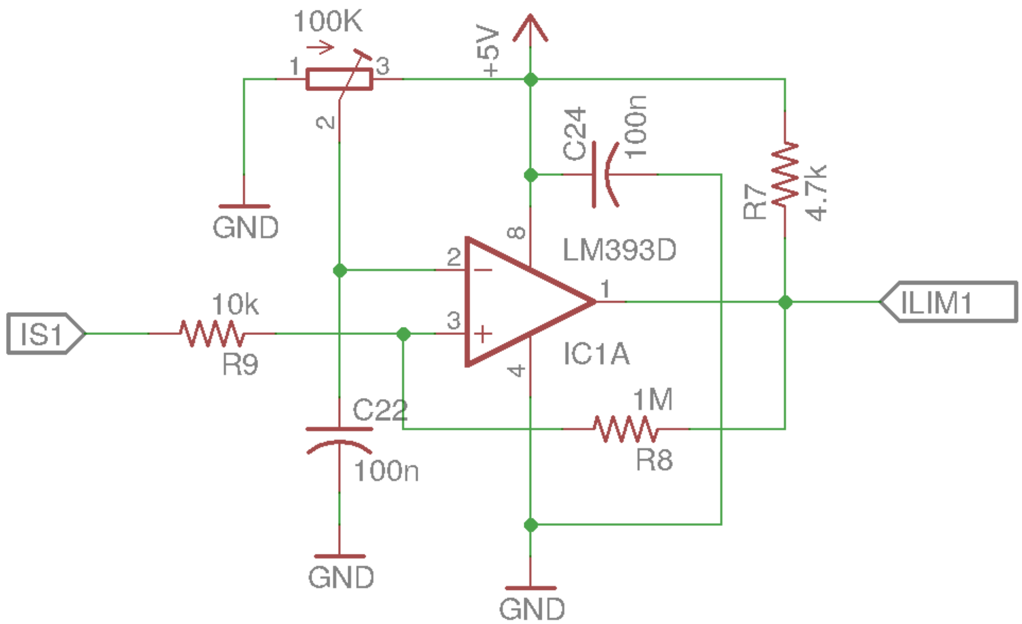


Figure 6: Current Sense Comparator Circuitry

2.5 Physical Design

Our final product was prototyped on a single hung 18"× 24" window. The window was surrounded by an L-shaped plywood frame with 5" on the top and 3" on one side, which we used to mount the main board, the rain sensor, and the temperature sensor boards.

On the inside of the window, we mounted the main board, IR sensors and emitters, one temperature sensor, and the comparator of the rain sensor module. We mounted the main board on the top to minimize the distance from the main board to the motor. We used three IR sensors for breakbeam obstruction detection, and evenly spaced these on the left side of the bottom half of the window. The corresponding IR LED was placed on the opposite side of the window, in line with the center IR sensor. One IR sensor and its corresponding emitter was used for bottom detection. We placed these on the bottom right corner of the top half of the window. They were arranged perpendicular to the plane of the window such that when the hung panel was anywhere but flush against the bottom, the beam would be broken. Lastly, the indoor temperature sensor and the comparator of the rain sensor were mounted on the plywood at the side of the window.

On the outside of the window, we mounted the exposed-lead PCB of the rain sensor and one temperature sensor. The exposed-lead PCB was mounted horizontally on an L-bracket on the plywood frame on the side of the window, and the outdoor temperature sensor was mounted directly underneath it.

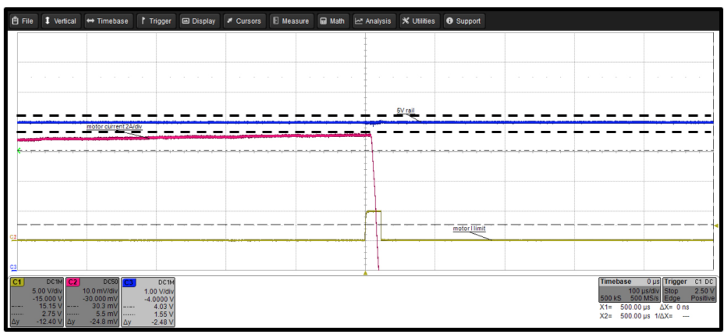
(a) Inside (b) Outside

Figure 7: Inside and Outside of Assembled Window

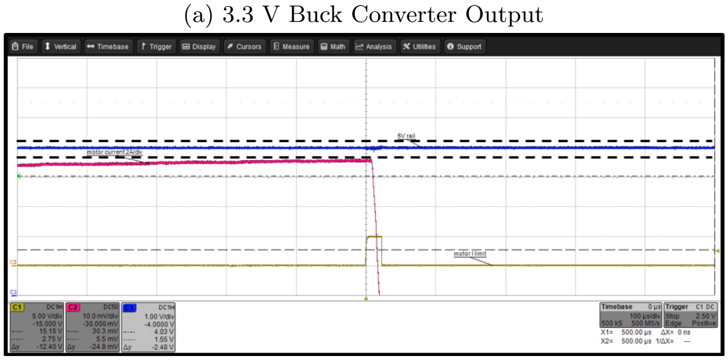
3 Verification

3.1 Power Module

To verify the functionality of our power module, we observed the voltage outputs of our three converters: our 120 VAC to 12 VDC converter, 12 V to 5 V buck converter, and our 12 V to 3.3 V buck converter. To verify our AC-DC converter, we probed the two output pins of the power supply and verified that there was a steady 12 V differential output coming from the converter. The requirements for our two buck converters were much more stringent. Our main requirement was to maintain a maximum 5% tolerance throughout all modes of operation. We can see the outputs of our buck converters in the worst-case scenario in Fig. 8. In this scenario, we jam the window to reach the motor's stall current, then reverse the direction of the window. The amount of ripple in the 3.3 V buck converter during a complete current reversal in the motor is well within our 5% tolerance specification, and the response of the 5 V converter is nearly the same.



(a) 3.3 V Buck Converter Output



(b) 5 V Buck Converter Output

Figure 8: Ripple in Buck Converters During Motor Current Flip. The top trace shows the output of the 3.3 V buck converter. The second trace shows the output of the motor current sense, which steadily rises until it jams, at which point the motor changes direction. The third trace shows the digital logic level output of the current sense comparator, which is active high when the motor jams. The two dotted black lines show the ±5% tolerance region.

3.2 Control Module

We performed verification of the control module by unit testing the subroutines that communicate with the various peripherals. This was especially important in testing the SPI communication with the ADC and in testing the software driver the the IR LED. See Section 3.3 for further discussion on validation of the SPI communication subroutine.

To test the overall state machine, we connected the Arduino Uno board to a computer via USB. We then used the internal 3.3 V, 5 V, and GND to simulate various inputs, such as those of the rain sensor or IR sensors, to trigger state transitions. We then had the state machine output its internal state variable to the Arduino IDE's serial monitor to verify the state transitions. After assembly, we observed the internal state of the state machine by observing the actions of the motor, as each state controlled the motor differently.

3.3 Sensor Module

First, we validated the IR sensor board by using a television remote control that is known to operate at 38 KHz, and observed the readout of the TSSP77038 on an oscilloscope. After we observed the active low output in the presence of an IR sensor, we tested it in conjunction with the IR LED and the software driver. We performed a distance test and a viewing angle test, ensuring that at the IR emitter's operating frequency of 15.6 KHz, we were able to achieve a transmission distance of 19" at a viewing angle of 20°, the maximum needed specifications for operation.

The most critical component to validate in the sensor module was the temperature board, as we needed to validate both the MCP9700A and the corresponding serial digital readout from the ADS7868. After board assembly, we validated the MCP9700A by using a digital multimeter and varying the ambient temperature by breathing on the sensor and applying an ice pack. Before reading the output of the ADS7868 on the microcontroller, we first observed the output on the oscilloscope, as in Fig. 9.

Fig. 9 shows that the ADC outputs binary `00111011`, equivalent to decimal 59. We convert this value to temperature as follows.

$$\frac{\frac{59}{255} \times 3300\text{mV} - 500\text{mV}}{10\text{mV}/°C} = 26°C \tag{13}$$

The values 255 and 3300mV come from the maximum output of the ADC and its corresponding voltage; 500mV and 10mV/°C are provided by [8] as the 0°C voltage offset and the voltage/temperature scale, respectively.

Lastly, we validated the rain sensor module by spraying water onto the exposed-lead PCB and reading the output on the microcontroller.

3.4 Motor Drive Module

The motor drive module had several requirements of the driver and the motor. Overall, the system was required to open or close the window in under 1 minute. This was verified in the final product by simply measuring the time it took to open and close the window. The motor and motor driver easily met this criterion.

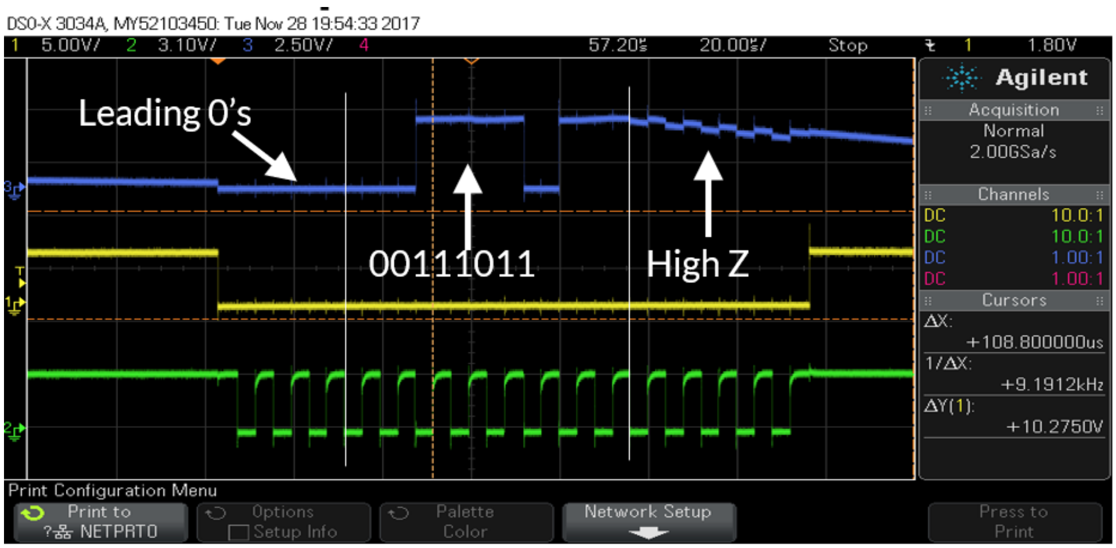


Figure 9: Oscilloscope Readout of the ADC Output. The top trace shows the ADC output. The second trace shows the slave select signal from the microcontroller, active low for sixteen clock cycles of the read instruction. The third trace shows the clock signal from the microcontroller.

The motor driver also needed the capability to drive the motor in both directions. The motor driver was bench tested with up to a 2.8A load. During the bench test, both driving directions were tested. The motor driver was also qualitatively verified to function in the final product, where the motor system reliably moved the window up and down a number of times.

The motor driver current sense was verified during the same bench test as the motor driver output. With a 2.8A load, the current sense comparator was verified to be finely tunable to trip with as little as 12mA of disturbance current, which is approximately a 0.4% change. During various tests with the final product, the current sense comparators functioned correctly and were able to detect window jams as well as hitting mechanical stops at the end of travel.

4 Cost

4.1 Labor Cost

For our labor, we estimate labor costs of \$35/hour. Over the ten weeks working on the project, we averaged 10 hours in the lab per person per week. Thus, our costs come out to be \$26,250 for the development of the Weather-Adaptive Window.

$$\frac{\$35}{hour} \times \frac{10hours}{week} \times 10weeks \times 3people \times 2.5 = \$26250 \tag{14}$$

For machine shop labor, we estimate labor costs of \$60/hour, and assume that this project required ten hours of work by the machine shop.

$$\frac{\$60}{hour} \times 10 hours = \$600 \tag{15}$$

4.2 Material Cost

In the following bill of materials, as the machine shop will be providing for us all materials, including window, opening mechanisms, and frame, we will subsume the cost of each item, assumed to total $75, under "Mechanical Components".

Component	Quantity	Unit Cost ($)	Total Cost ($)
Main Board			
0.022μF Capacitor	2	0.017	0.03
22μF Capacitor	4	0.182	0.73
10μF Capcaitor	1	0.19	0.19
0.1μF Capacitor	8	0.02	0.16
18pF Capacitor	2	0.031	0.06
0.22μF Capacitor	4	0.076	0.30
100μF Capacitor	1	0.24	0.24
Schottky Diode (BAS40)	2	0.27	0.54
Dual Comparator (LM39MFS)	1	0.35	0.35
DC-DC Regulator (TS30013-M033)	1	1.19	1.19
DC-DC Regulator (TS30013-M050)	1	1.19	1.19
Motor Driver (BTN8962TA)	2	5.11	10.22
Microcontroller (ATmega328P)	1	2.00	2.00
16MHz Crystal (ABLS-16.000MHZ)	1	0.25	0.25
9 Pin D-Sub Connector	1	0.78	0.78
15 Pin D-Sub Connector	2	1.36	2.72
8 Pin Female Header	1	0.67	1.34
3.6KΩ Resistor	2	0.015	0.03
4.7KΩ Resistor	2	0.011	0.02
1MΩ Resistor	2	0.011	0.02
10KΩ Resistor	8	0.011	0.09
100KΩ Potentiometer	2	1.50	3.00
4.7μH Inductor	2	1.13	2.26
Main PCB	1	12.60	12.60
IR Sensor Board			
IR Sensor (TSSP77038)	4	1.74	6.96
ESD Diode (GSOT05C)	4	0.314	1.26
330Ω Resistor	4	0.026	0.10
1KΩ Resistor	4	0.014	0.06
2.2μF Capacitor	4	0.133	0.53
IR PCB	4	6.50	26.00
Temperature Sensor Board			
Temperature Sensor (MCP9700A)	2	0.33	0.66
ADC (ADS7868)	2	1.50	3.00
Reference (REF3133)	2	2.71	5.42
0.1uF Capacitor	2	0.01	0.02
0.047uF Capacitor	2	0.09	0.18
Temperature PCB	2	12.90	25.80
Standalone Components			
40 Count Male Headers	1	0.66	0.66
Motor	1	24.95	24.95
Wi-fi Module (ESP8266-01)	1	6.95	6.95
Rain Sensor	1	7.95	7.95
Wall Adapter	1	11.88	11.88
Mechanical Components	N/A	N/A	75.00
Total	N/A	N/A	263.69

Table 1: Bill of Materials

Thus, our total material costs come out to $263.69.

5 Conclusion

5.1 Accomplishments

In the end, we were able to design the window to close at the speed we desired. We were also correctly implement the functionality of the rain sensor and infrared sensors such that the window would close in the presence of rain but also respond intelligently to the presence of an obstruction. Also accomplished was the jam detection functionality, which meant that the window would be able to sense when it reaches the top or bottom and any objects that might prevent the window from closing. We were also able to correctly implement the state machine, which meant that the window could operate on its own. By knowing what state it was in, the window would transition on its own, correctly controlling the motor along the way.

5.2 Uncertainties

There were some uncertainties in the debugging phase of our project after the assembly of our window. The first concerns our 120 VAC to 12 VDC converter, which is rated for 8 A. During testing, we realized that the stall current of the motor was higher than expected: though the motor is rated for 5 A stall current, we found that when the window jammed for an extended period of time, the stall current could reach 8 A. In this case, the wall-adapter's internal over-current protection would shut down our power supply.

The second uncertainty is with the SPI communication between the microcontroller and the ADC in the sensor module. During unit testing, we were able to properly read out the temperature, but after assembly, the temperature module did not output the expected values. We believe that this is due to our placement of the temperature board far from the main board. The long wires begin to act as antennas and we witnessed significant crosstalk between the SPI lines, which showed up on the oscilloscope. In particular, we believe that the ADC may have been reading crosstalk from the SPI clock line as multiple slave select signals and never finished its 8-bit serial transmission before beginning another write operation.

5.3 Ethics and Safety

As with any automated system, there are safety concerns. In this case, the main safety concern is with closing the door on any living being, whether it be an unaware person or animal that may be occupying the window space as it attempts to close itself. To prevent harm to anything in the way of the window, an array of infrared sensors will be used as object detection. These will be used as logical input to the microcontroller, and the output of these sensors will be of the utmost importance when controlling the motor - even more important than the weather indicators themselves. Also, infrared sensors will be used to gauge when the door has closed in addition with current sense at our motor. This prevents the motor from placing excessive stress on the window housing which could lead to shortened durability of the mechanism.

Another concern arises when dealing with electronics in an outdoors context. Our rain and temperature sensors will have to be outward facing. In order to ensure that only our sensors are facing on the outside, we will have to keep the bulk of the circuitry on the side of the window that faces toward the room. The sensors will likely need some sort of housing to ensure that none of the rain or humidity can reach the circuitry, which would cause undesired shorts. These shorts can lead to explosive results, which could cause harm to the end user.

In the lab, working with power conversion is always a concern. In a situation in which components or converters may not be properly connected, some parts could be overloaded and explode. In addition, when working with motors, there's always the concern that the motor could behave unexpectedly it isn't controlled properly which could lead to bodily harm.

A concern that was raised during the mock presentation of our project was the issue of wireless security. As with any wireless communication, there exists the risk of someone hijacking the communication, which could pose a significant security risk to the user. If an intruder is able to control the window at his or her own will, that could compromise the safety of the user's home or building. Therefore, with the implementation of wireless functionality must come wireless security.

Ethically, we believe that there are few concerns with our project. One promise that we made over the course of our project is disclosing various aspects of our project that might endanger people or the world that people live in, which is in accordance with the first point of the IEEE Code of Ethics, "...to disclose promptly factors that might endanger the public or the environment," [3]. In addition, we have many prevention measures in place to ensure that we follow item nine on the Code of Ethics, "to avoid injuring others...by false or malicious action," [3]. Apart from these concerns, we believe that there are few aspects of our project that might be of ethical concern.

5.4 Future Work

In the future, we hope to achieve Wi-Fi functionality. Though we were able to get basic functions of the ESP8266 working, we were never able to get to a point where it could do enough to operate the window. We also realized along the way that many of our issues understanding the functionality of the ESP8266 could have been alleviated through the choice of an easier-to-use Wi-Fi module. As mentioned in the ethics and safety portion, this wireless communication should also come with some sort of security functionality.

We also hope to work out the bugs in the SPI communication, likely starting by moving the temperature sensors closer to the microcontroller. We originally wanted the window to open and close itself based on a user-programmable threshold but were not able to achieve this.

In addition to meeting our original goals, there are other features we would like to add to the window design that address the concerns posed in Section 5.3. Our current design does not provide housing for the outdoor temperature sensor, nor does it provide a security feature for the Wi-Fi capability, as our aim in this project was foremost to prototype basic functionality of the window. Future models would certainly have to have solutions for these issues.

A Schematics and Layouts

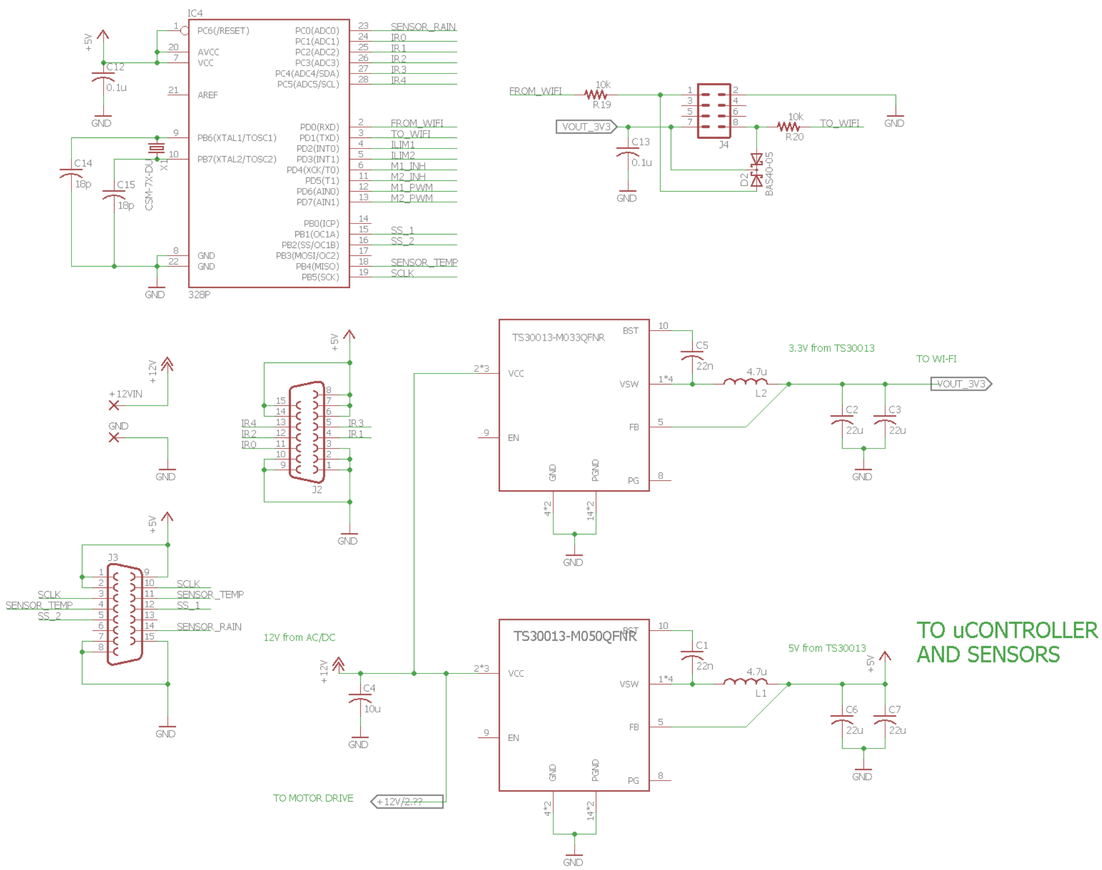


Figure 10: Main Board Schematic Page 1: Microcontroller, Wi-Fi Module, Buck Converters

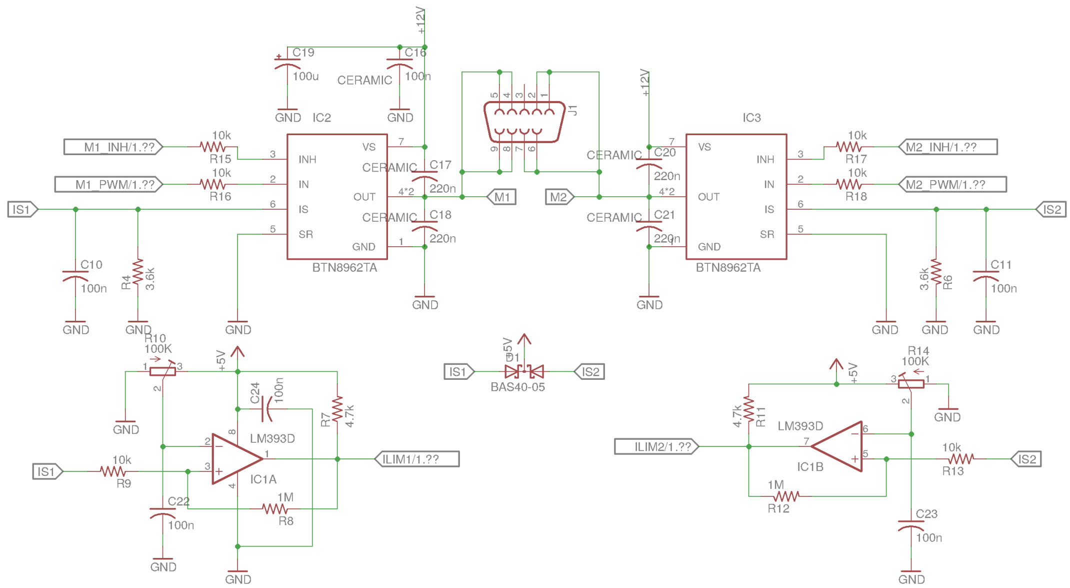


Figure 11: Main Board Schematic Page 2: Motor Drivers, Current Sense Comparators

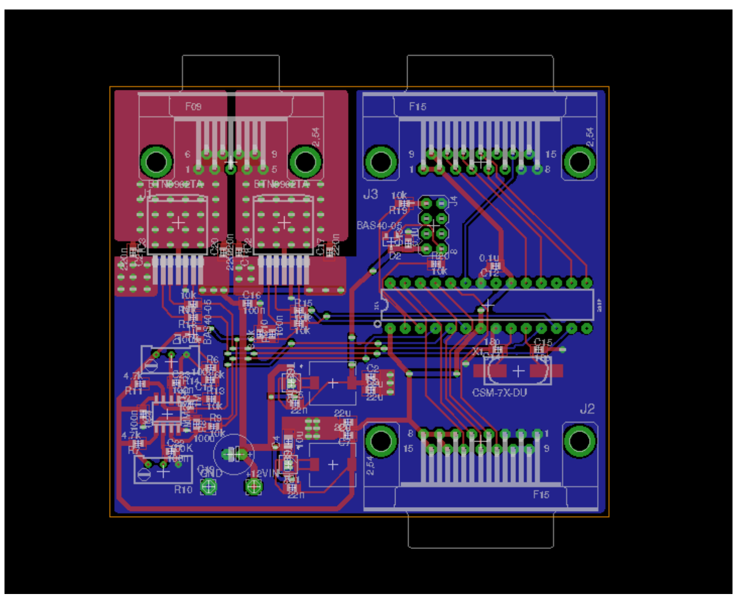


Figure 12: Main Board PCB Layout

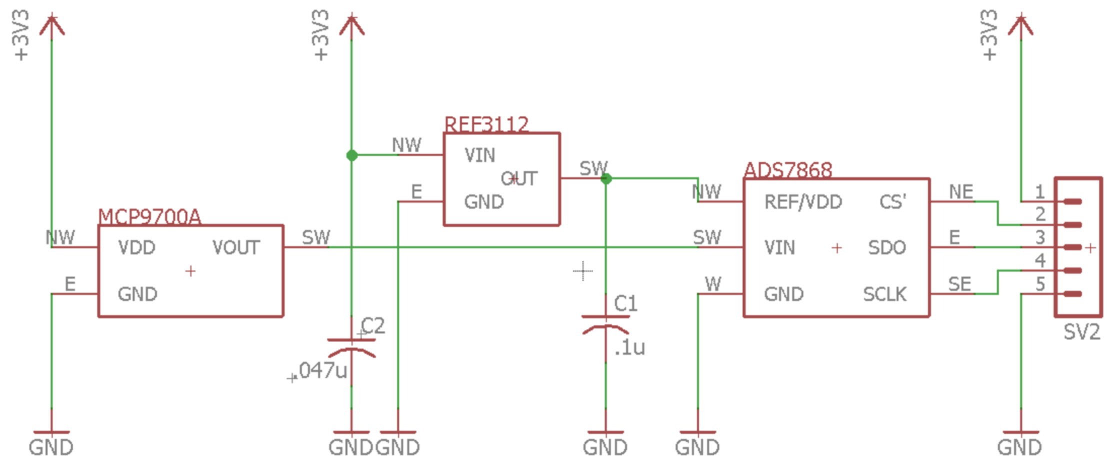


Figure 13: Temperature Sensor Schematic

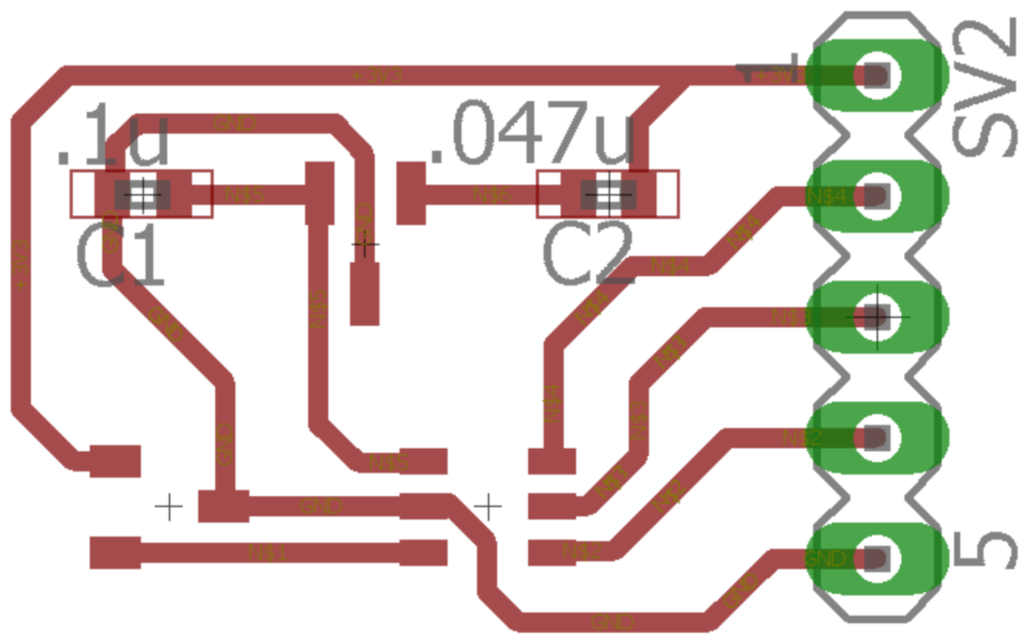


Figure 14: Temperature Sensor PCB Layout

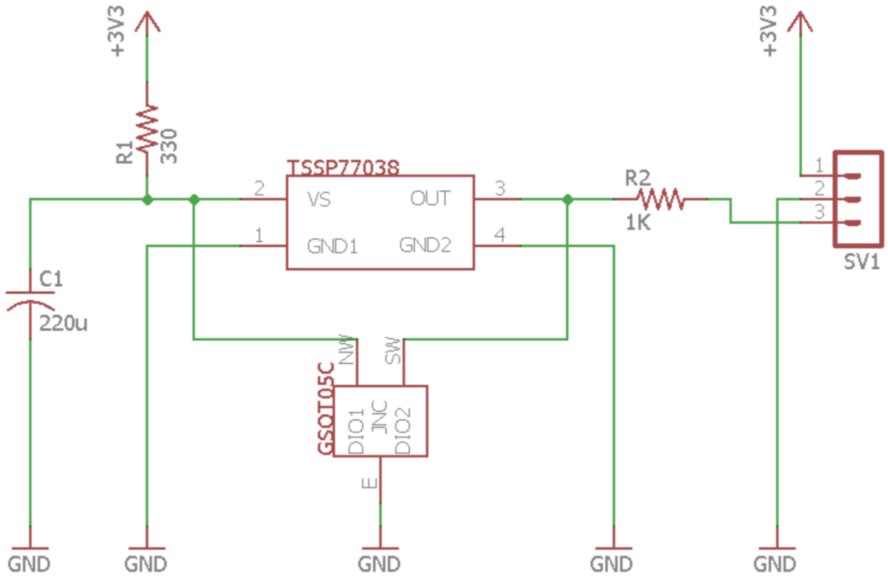


Figure 15: Infrared Sensor Schematic

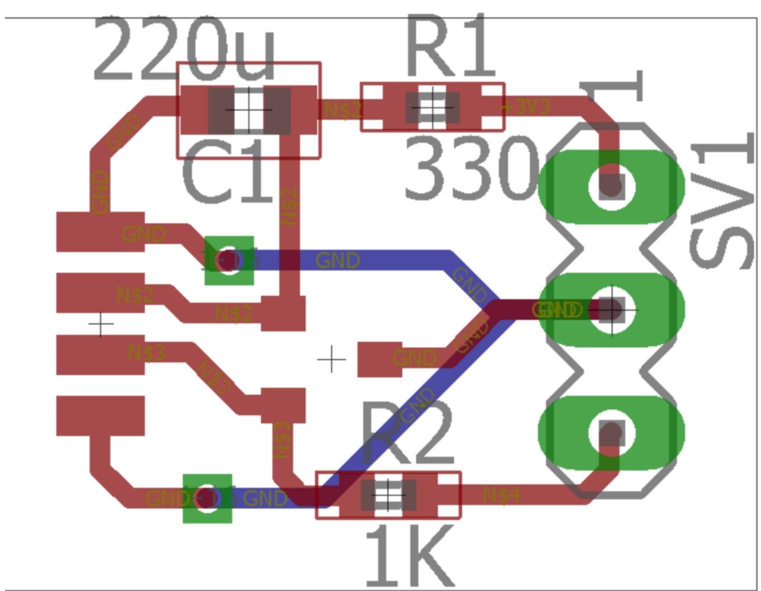


Figure 16: Infrared Sensor PCB Layout

B Requirements and Verification

Requirements	Validation	Points
Convert from 120V 60Hz AC power to 11-13V DC with 80% efficiency	1. Insert adapter into wall 2. Use breakout board to connect output to breadboard 3. Use Digital Multimeter (DMM) to probe output	1

Table 2: Requirements and Validation: Wall Adapter

Requirements	Validation	Points
Reliably step down voltage from wall adapter output, 11-13V, to low voltage power rails, a 4.5-5.5V rail and a 2.5-3.3V rail	1. Connect inputs to a power supply with known power output 2. Use a DMM to probe the outputs to ensure that they fall in the correct range.	2
Maintain 85% efficiency while stepping down from 12V at 0.5A.	1. Connect inputs to a power supply with known power output 2. Connect outputs to breadboarded circuit drawing 0.5A at desired output voltage 3. Use DMM to probe the outputs to ensure that efficiency is met.	1
Maintain 5% tolerance through powered component startup, shutdown, and operation	1. Connect the outputs of the regulators to other components of our circuitry 2. Use an oscilloscope to probe the output of the regulators through startup and shutdown to ensure that tolerance is met.	1

Table 3: Requirements and Validation: DC-DC Step Down Regulators

Requirements	Validation	Points
Receive 6-8 bit serial digital input from the temperature sensor via an ADC and digitally compare it to a user-preset value, input from the Wi-Fi peripheral	1. Write test embedded software with handshakes between ADC and microcontroller 2. With a DMM, compare the outputs from the ADC to the readouts from the microcontroller. 3. Check that the microcontroller raises the correct flag when a temperature threshold is crossed	2
Rapidly poll the input from the IR sensors, at least 20 Hz, when closing, to determine the position of the window and to detect any obstructions	1. Write test embedded software with a polling flag, outputted to one GPIO pin 2. Use the oscilloscope to ensure that the flag is activated at least at 20 Hz	2
Implement closed-loop control logic to output forward, reverse, or halt instructions to the motor driver	1. Simulate different artificial responses from each sensor using the power generator to probe the GPIO pins of the microcontroller 2. Use the oscilloscope to probe the GPIO that are pins to be connected to the motor driver	15

Table 4: Requirements and Validation: Microcontroller

Requirements	Validation	Points
Have at least 15dBm output power at 2.4GHz, similar to a laptop's wireless local area network chip	1. Program ESP8266 to broadcast at 2.4GHz 2. Broadcast a continuous wave as well as pulsed waves 3. Use an RF power meter to measure signal from antenna	2
Transmit signal to user smartphone from a distance of at least 10m	1. Write C++ code for ESP8266 to send a text to a phone 2. Use a tape measure to mark off distances spaced 1m apart 3. Send messages to user at each of the marked distances	6

Table 5: Requirements and Validation: Wi-Fi module

Requirements	Validation	Points
Output a digital binary value to the microcontroller in the presence of rain	1. Plug connectors from rain sensor into breadboard 2. Using a pipette/eyedropper, place varying levels of water around the rain sensor 3. Use an DMM to probe the outputs of the rain sensor	1

Table 6: Requirements and Validation: Rain Sensor

Requirements	Validation	Points
Sensor monitors temperature of ambient environment with accuracy of $\pm 1°C$	1. Characterize temperature sensor in different controlled-temperature environments, e.g. refrigerator 2. Probe analog outputs using hand-held DMM 3. Compare readouts to traditional thermometer or to known ambient temperature	2
ADC has at least 6 bits of resolution to provide at least 128°C range while satisfying accuracy requirement	1. Simulate output of temperature sensor using power supply to sweep voltage of input to ADC 2. Use oscilloscope to read the serial output from the ADC	1

Table 7: Requirements and Validation: Temperature Sensor

Requirements	Validation	Points
Have at least a 18 inch range to span the reasonable width and height of a window	1. Probe the output of the IR sensor with an oscilloscope 2. Gradually sweep objects of varying reflectance from 1 to 18 inches and record digital readout 3. Compare reading at 18 inches with reading from effective infinity distance	4

Table 8: Requirements and Validation: IR sensors

Requirements	Validation	Points
Have current measurement capability for motor current feedback purposes, and can stop motor operation when supplying 3A	1. Run the motor with different loads by hanging a weight on the motor 2. Probe output of motor driver with oscilloscope and record readout 3. Artificially stall the motor and record readings from motor driver	2

Table 9: Requirements and Validation: Motor Driver

Requirements	Validation	Points
Motor must be able to open and close the window one foot in one minute with gear reduction from the lead screw	1. Use yardstick to mark off distances on the window frame 2. Time the opening and closing operations with a stopwatch	8

Table 10: Requirements and Validation: Motor

C Tolerance Analysis

A key aspect of a self closing window is that it is completely safe. In order to achieve a safe design, infrared obstruction detection will be used to ensure no objects are in the window while it is closing. However, being a window, the sun provides a large source of noise to our measurement, with the capability of completely drowning out the signal we are looking for (that of an infrared emitter). There is a simple solution to this problem: The sun can be considered a DC source of infrared. Its light is mostly unvarying, so the infrared measurement will appear to have an offset. By pulsing our infrared source at high frequency, 56 kHz, we can ensure that our signal is well outside the bandwidth of the sun's infrared. Then, the measurement from the infrared sensor will be high pass filtered to remove the noise introduced by the sun. This filter can be a simple first order R-C filter. The cutoff frequency of a filter such as this is calculated by $f_{cutoff} = \frac{1}{2\pi RC}$.

The cutoff of this filter must below the frequency that the IR emitter pulse at, but significantly above the low frequencies of the sun's IR in order to maximize the removal of the noise. Resistor tolerances are typically between $\pm1\%$ and $\pm10\%$, while typical capacitor tolerances are larger, $\pm5\%$ to $\pm10\%$.

The cutoff frequency for a single pole RC filter like this can therefore vary by as much as approximately $(105\% * 110\%) - 100\% = 16\%$ for a 5% resistor and a 10% capacitor, due to the multiplicative effect of component tolerances. From the low end tolerance to the high end tolerance, this is as much as a 32% swing. Part of the solution is to use components with tighter tolerances. However, tighter tolerance components cost slightly more than they're loose tolerance counterparts. Another solution to the issue is to ensure that the signal (our pulsing IR emitter) is above the upper cutoff, even for the highest possible cutoff frequency. This means that our nominal cutoff must be less than $F_{c,max}/1.16$. Using 38 Khz for $F_{c,max}$, we get a nominal cutoff of 32.7 kHz, which also results in a minimum cutoff of 28.7 kHz. This is a large spread, but as long as our minimum cutoff is far above the frequency of the sun's IR, this still provides a significant attenuation of the sun's IR.

References

[1] Business Wire, "Research and Markets: Global Home Automation and Control Market 2014-2020 - Lighting Control, Security & Access Control, HVAC Control Analysis of the $5.77 Billion Industry," Berkshire Hathaway, 19 January 2015. [Online]. Available: http://www.businesswire.com/news/home/20150119005490/en/Research-Markets-Global-Home-Automation-Control-Market. [Accessed 21 September 2017].

[2] Nest, "Meet the Nest Learning Thermostat," 2017. [Online]. Available: https://nest.com/thermostats/nest-learning-thermostat/overview/. [Accessed 21 September 2017].

[3] IEEE, "IEEE Code of Ethics," IEEE, 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed 20 September 2017].

[4] eBay Inc., "AC DC Power Supply Adapter Transformer 12V 2/3/5/6/8/10A for 5050 3528 LED Strip," [Online]. Available: https://www.ebay.com/itm/222351684139. [Accessed 7 November 2017].

[5] Semtech Corporation, "High Efficiency Current-Mode Synchronous Buck DC/DC Converter," TS30013 datasheet, Aug. 2016.

[6] Atmel Corporation, "Atmel 8-Bit Microcontrontroller with In-System Programmable Flash," ATmega328P datasheet, 2015.

[7] Expressif Systems, "Highly Integrated Wi-Fi SoC Solution," ESP8266 datasheet, 2017.

[8] Microchip Technology, "Low-Power Linear Active Thermistor™ ICs," MCP9700a datasheet, Apr. 2014.

[9] Texas Instruments, "Micro-Power, Minature Analog-To-Digital Converter With Serial Interface," ADS7868 datasheet, June 2005.

[10] Texas Instruments, "Ref31xx 15ppm/°C Maximum, 100-μA, SOT-23 Series Voltage Reference," REF3133 datasheet, Dec. 2003 [Revised Mar. 2016].

[11] Vishay Semiconductors, "IR Sensor Module for Reflective Sensor, Light Barrier, and Fast Proximity Applications," TSSP77038 datasheet, Feb. 2017 [Revised Aug. 2017].

[12] Ledtech, "5.00MM Round Infrared Lamp," UT188X-81-940 datasheet, July 2002.

[13] Vishay Semiconductors, "Two-Line ESD Protection in SOT-23," GSOT03C datasheet, Mar. 2016.

[14] Infineon Technologies, "High Current PN Half Bridge, NovalithIC™," BTN8962TA datasheet, May 2013.